Presto! Distilling Steps and Layers for Accelerating Music Generation

[1]UC – San Diego

Zachary Novack[1,2], Ge Zhu[2], Jonah Casebeer[2], Julian McAuley[1], Taylor Berg-Kirkpatrick[1], Nicholas J. Bryan[2]

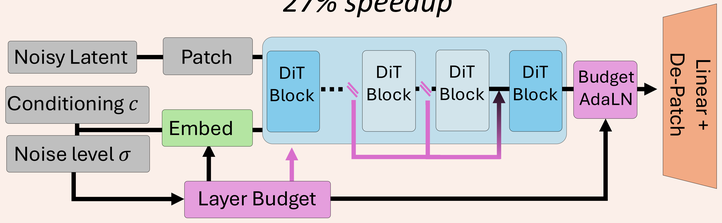
[2]Adobe Research

Project Personal

How can we accelerate Text-to-Music Diffusion models for real-time, high-quality generation?

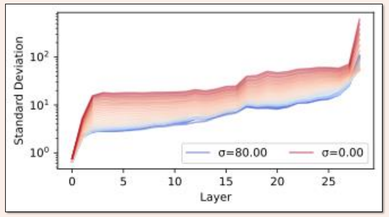
Presto-L: Reducing *Cost* per Step

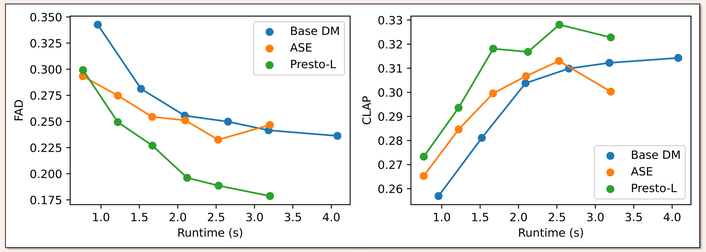
Presto-L: *SOTA layer dropping method for standard DMs, 27% speedup*



Finetune to **drop layers** based on **noise level**:

- Early steps **(high noise)** are easier -> drop more layers
- **Explicit budget conditioning** (global embed + final AdaLN)
- Reroute to **final DiT layer** for variance preservation



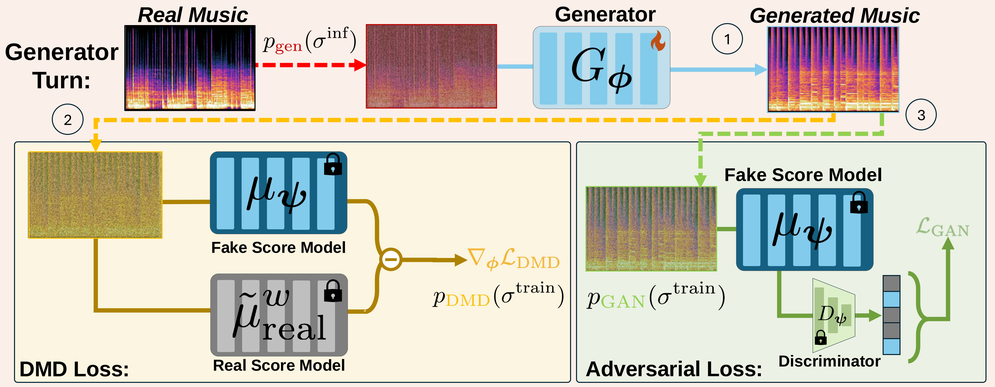


Against base DM and layer-dropping baseline, **Presto-L** is faster and *improves* performance (multi-task specialization)

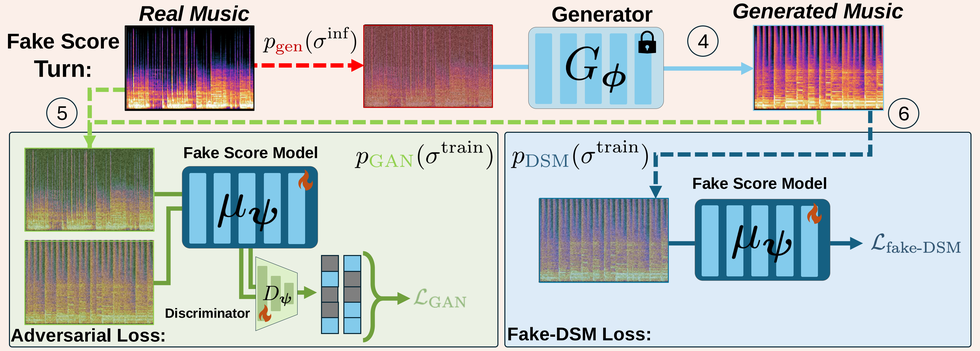
Presto-S: Reducing *Number* of Steps

Presto-S: *1st Adversarial TTM diffusion distillation, 15X speedup*

- Extends Distribution Matching Distillation (DMD) to score-based models
- Choice of noise distribution (train vs. inf) is **critical** for SOTA performance



Push generator close to real data "in distribution" (KL) ***Make generator fool discriminator (GAN)***

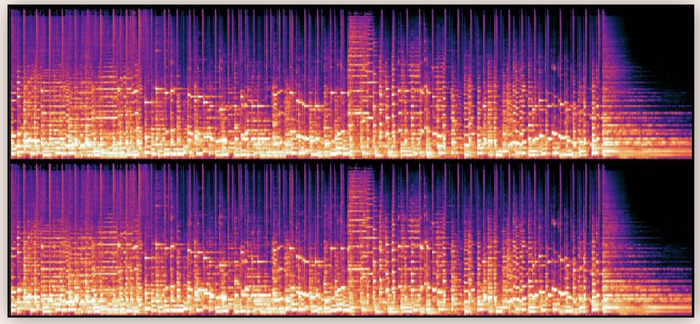


Make discriminator detect generator (GAN) ***Make fake SM track generator's score (diffusion)***

Presto-LS: Maximum Speed!

Presto-LS: *1st combined layer-step method, 18X speedup*

- Drop layers, *then* steps
- Use full-rank auxiliary models for Presto-S step
- Drop fewer layers 😭



SOTA performance: better **quality** than base DM, **faster** than other distillation methods, more **diverse**

TLDR: 32s of Mono/Stereo 44.1kHz music in *0.23/0.43* seconds

Check out the paper for more info!

Further Analysis:

- Continuous vs. discrete DMD
- Subjective Listening Study
- Layer dropping ablations
- Presto-LS ablations

Extra Use Cases:

- 45s **CPU** latency
- Adaptive-Step Schedule
- Inference-Time Scaling